

## **AMENDMENTS TO THE SPECIFICATION**

Please amend the specification by replacing the section titled "Brief Description of the Drawings," located on pages 6-9 with the following section:

### **BRIEF DESCRIPTION OF THE DRAWINGS**

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an implementation of the invention and, together with the description, serve to explain the advantages and principles of the invention. In the drawings,

FIG. 1 depicts a conventional software development tool;

FIG. 2 depicts an overview of a software development tool in accordance with the present invention;

FIG. 3 depicts a data structure of the language-neutral representation created by the software development tool of FIG. 2;

FIG. 4 depicts representative source code;

FIG. 5 depicts the data structure of the language-neutral representation of the source code of FIG. 4;

FIG. 6 depicts a data processing system suitable for practicing the present invention;

FIG. 7 depicts an architectural overview of the software development tool of FIG. 2;

FIG. 8A depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a list of predefined criteria which the software development tool checks in the source code;

FIG. 8B depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays the definition of the

SSK/mem

criteria which the software development tool checks in the source code, and an example of source code which does not conform to the criteria;

FIG. 8C depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays an example of source code which conforms to the criteria which the software development tool checks in the source code;

FIG. 9 depicts a flow diagram of the steps performed by the software development tool depicted in FIG. 2;

FIGS. 10A and 10B depict a flow diagram illustrating the update model step of FIG. 9;

FIG. 11 depicts a flow diagram of the steps performed by the software development tool in FIG. 2 when creating a class;

FIG. 12 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a use case diagram of source code;

FIG. 13 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays both a class diagram and a textual view of source code;

FIG. 14 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a sequence diagram of source code;

FIG. 15 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a collaboration diagram of source code;

FIG. 16 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a statechart diagram of source code;

FIG. 17 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays an activity diagram of source code;

FIG. 18 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a component diagram of source code;

FIG. 19 depicts a user interface displayed by the software development tool depicted in FIG. 2, where the user interface displays a deployment diagram of source code;

FIG. 20 depicts an exemplary data processing system in which the improved software development tool depicted in FIG. 2 may operate;

FIGS. 21A-C depicts a flow diagram illustrating an exemplary process performed by the software development tool for generating source code for an object-oriented element from a data structure in a database;

FIG. 22 depicts a user interface displayed by the software development tool for receiving a request to access a database;

FIG. 23 depicts a user interface displayed by the software development tool for receiving access information for the database;

FIG. 24 depicts a user interface displayed by the software development tool, where the user interface displays an identification of each data structure stored in the database;

FIG. 25 depicts a user interface displayed by the software development tool, where the user interface displays an indication of a type of object-oriented element to form from the database;

FIG. 26 depicts a user interface displayed by the software development tool, where the user interface displays source code for a class that is generated from a selected data structure in the database;

FIG. 27 depicts a user interface displayed by the software development tool depicted, where the user interface displays source code for a Enterprise JavaBean.TM. that is generated from a selected data structure in the database;

FIG. 28 displays the source code for the Enterprise JavaBean.TM. in FIG. 27 that is generated by the software development tool;

FIGS. 29A-B depict a flow diagram illustrating an exemplary process performed by the software development tool for updating an object-oriented element from a related data structure in a database;

FIG. 30 depicts a user interface displayed by the software development tool, where the user interface displays source code for a distributed computing component that is updated from the related data structure in the database;

FIGS. 31A-B depict a flow diagram illustrating an exemplary process performed by the software development tool for forming a data structure in a database from source code of an object-oriented element.

FIG. 32 depicts a user interface displayed by the software development tool for receiving a request to form a data structure in a database;

FIG. 33 depicts a user interface displayed by the software development tool for receiving an indication of an object-oriented element to use to form the data structure;

FIG. 34 depicts a user interface displayed by the software development tool for receiving access information for the database and for receiving a request to generate the data structure in the database;

FIG. 35 depicts a user interface displayed by the software development tool depicted, where the user interface displays a location for storing the data structure formed from source code of the object-oriented element; **and**

FIGS. 36A-B depict a flow diagram illustrating an exemplary process performed by the software development tool for updating a data structure in a database from a related object-oriented element; and

FIG. 37 depicts an Enterprise JavaBean™ server interacting with other systems.

Reference will now be made in detail to the description of the invention as illustrated in the drawings. While the invention will be described in connection with these drawings, there is no intent to limit it to the embodiment or embodiments disclosed therein. On the contrary, the intent is to cover all alternatives, modifications, and equivalents included within the spirit and scope of the invention as defined by the appended claims.

Please amend the paragraph spanning pages 28-29 in the specification as follows:

An example of an EJB representing a real-world object (e.g., a customer of an enterprise) is shown below depicted in FIG. 37. Each record in the "Customer Table" Customer Table 3705 found in the database 3710 is accessible by the example "Customer EJB" Customer EJB 3715 via the attribute fields 3720 of the Customer Table 3705 (i.e., "ID", "Name", "Social Security Number", and "Age" are shown) that are mapped to attributes 3725 in the code corresponding to the EJB. Assuming the "Client Application" Client Application 3730 needs to determine whether the person named "Jones" qualifies for a senior citizen discount on a purchase order, the "Customer EJB" Customer EJB 3715 may be invoked to answer the question by performing the necessary business logic operation which may involve ascertaining the "age" for "Jones" from the "Customer Table" Customer Table 3705 and determining if

SSK/mem

the "age" for "Jones" is over 50 before reporting back the answer to the "**Client Application.**" **Client Application 3730.**

Please delete the illustration located on page 29 of the specification.

SSK/mem